REMARKS

This Amendment is in response to the Office Action dated June 21, 2005 in which claims 1-16 were initially rejected and claims 17 and 28 were withdrawn from consideration as being directed to a non-elected species. Applicants respectfully request reconsideration and allowance of claims 1-16 in view of the above-amendments and the following remarks.

I. SPECIFICATION

The disclosure was objected to because of a typographical error on page 5, line 13 in the spelling of the term "uncoded".

However, this typographical error was corrected in the previous Amendment, filed May 11, 2005. Consideration of the previous Amendment is respectfully requested.

II. CLAIM OBJECTIONS

A. **Claims 2, 10 and 15**

Claims 2 and 10 were objected to since the Examiner prefers the phrase --the next data word in the sequence of data words-- over the original phrase "a next subsequent one of the data words in the sequence of data words."

The original language was intended to positively introduce the new term, "a next subsequent data word" with clear and explicit antecedent basis rather than relying on inherency. Each version is equally definite. Although Applicants believe the original phrase is sufficient, Applicants are willing to amend the phrase according to the Examiner's preferred language.

Claim 15 was objected to because of the extra term, "adding" in line 2. Accordingly, this term is cancelled.

With these Amendments, Applicants respectfully request that these claim objections be withdrawn.

B. **Claims 6, 7 and 14**

Claims 6 and 14 were objected to as being of improper dependent form for allegedly failing to further limit the subject

matter of a previous claim. Specifically, the Office Action suggests that the limitations found in claim 6 are substantially the same as lines 9-11 of independent claim 1.

However, claim 6 requires, "selectively complementing the binary symbol with the data word." An example of this embodiment is shown in FIG. 4, where step 403 appends the binary symbol to data word w to form cw, and cw is selectively complemented in step 405. Independent claim 1 does not require complementing the binary symbol with the data word. Claim 6 thus has a different scope than claim 1 and is in a proper dependent form. A similar analysis can be applied to claim 14.

Claim 7 was objected to as being a substantial duplicate of claim 4. However, claim 7 depends from claim 6, which has a different scope than independent claim 1. In contrast, claim 4 depends directly from independent claim 1. Therefore, claim 4 does not include the additional limitations of dependent claim 6. As a result, claim 7 is clearly not a substantial duplicate of claim 4 since it has a different scope.

Applicants therefore respectfully request that the objection to claims 6, 7 and 14 be withdrawn.

III. CLAIM REJECTIONS UNDER §112

Claims 1-16 were rejected under §112, first paragraph, as allegedly failing to comply with the enablement requirement. The Office Action provides a list of questions regarding the operation of some of the embodiments disclosed in the specification. As shown below, the answers to these questions can be found in the original specification.

Applicants' attorney appreciates that the Examiner is given very little time in which to review each application being examined and to perform a prior art search. Given this time restriction, questions will no doubt arise from time to time. The following discussion provides answers to each category of questions, in order.

As a background to this discussion, enablement is analyzed commensurate with the scope of the claims. A claim is enabled if the specification enables a person of ordinary skill in the art to make and use the invention that is being claimed without undue experimentation. Based on these criteria, all claims 1-16 are fully enabled by the specification.

In addition, not all of the Examiner's questions relate to elements appearing in all the claims. Some of these questions may therefore be irrelevant to the invention being claimed. In any case, all questions are addressed below.

A. **DC-Free and Bounded RDS**

Paragraphs 2 and 3 on page 4 of the Office Action discusses DC-free codes, a bounded RDS and limiting the RDS value.

In paragraph 4, the Examiner asked, "How does the step limit the RDS value?" More specifically, the Examiner poses the question of how the step of appending a binary "1" if the string's current RDS has the same sign as the RDS of the block, and appending a binary "0" if the signs differ would limit the RDS value. For example, the Examiner suggests that if the string had an excess of binary "1" values, it would have a positive sign and in such case if the block also had a surplus of "1" bits then adding another binary "1" would seem to increase the RDS rather than limit it within a tight bound.

Toward the end of line 1 in that paragraph, the Examiner failed to include the word "complementing" after "involves" and before "appending a binary 1". It is this complementing operation that keeps the RDS bounded. Applicant mentioned that the bits of the data word complemented in the scenario that the Examiner mentioned. (Specification, page 6, line 10.)

Example: Let the current value of R, i.e., the value of the accumulated RDS of all the (n+1)-bit code words so far, be

positive. Now, if the computed running-digital-sum for the next n-bit data word equals a positive number m, then by complementing all the bits in that word and appending a "1" would make the running-digital-sum of that word equal -m + 1. Therefore, the accumulated RDS after this new word equals R - m + 1, which is less than R.

If we do not complement the bits, then, as Examiner thought correctly, the net RDS will equal R + m + 1, which is greater than R. The complementing operation (i.e., turning 1's into 0's and 0's into 1's) saves us from increasing the RDS value. The same scenario applies if R is negative and m is negative.

This scenerio matches the criteria (R>0 and Ri>=0) defined in step 203 of FIG. 2 in which the data word is complemented. Thus, the Examiner's concerns regarding this hypothetical seem unwarranted.

B. **WHAT HAPPENS IF THERE ARE THE SAME NUMBER OF "1" BITS AND "0" BITS?**

On page 5, paragraph 1 of the Office Action, the Examiner questions what happens if there are the same number of "1" bits and "0" bits. Initially, the RDS of the codeword sequence is "0". The Examiner concludes that a zero is neither positive or negative and thus there would be no need to add an extra bit since the string is already balanced.

The fact that +0 and -0 are the same does not mean that zero can have no sign. It just means that the sign makes no difference. For example, see the attached copy of a January 9, 2002, posting on the web page,

http://mathforum.org/library/drmath/view/57215.html

in which "Dr. Math" states that zero can in fact have a sign.

Initially, we would assume that the initial accumulated RDS value equals zero. In other words, we start off with R = 0. The algorithm does not require any positive zero and negative

zero concepts.

Let us say the accumulated RDS value thus far equals R. (This covers the special case of R = 0 also.) Suppose that the next n-bit data word D is balanced, i.e., running-digital-sum, Ri, for that word equals 0. We stated in lines 17-20 on page 6 of the description how to handle such a scenario (Ri=0 can be regarded as a negative RDS value). Basically, we propose to simply append a "0" to the data word D to form the code word for D. This makes the running-digital-sum of that particular code word equal "-1". So, the net RDS after joining that code word to the code word stream built thus far equals R-1. (In the special situation where the cumulative R=0, the new cumulative value then equals -1). Afterwards, the algorithm would proceed normally.

In alternative embodiment, the specification mentions in lines 19-20 of page 6 that we could complement the data word D and append a "1" to it. Then the cumulative RDS after that operation will be R + 1. Either of these methods is OK.

A person of ordinary skill in the art could therefore easily make and use the invention that is being claimed without undue experimentation.

The Examiner also stated on page 5, lines 5-6 that, in the hypothetical, "there would be no need to add an extra bit." This statement is incorrect. In the embodiments disclosed in FIGS. 2 and 4, for example, a binary symbol (either a "1" or a "0") is appended to the data word in all cases. Thus, all codewords at this stage in the overall encoding sequence have the same number of bits, which makes it easier to decode the codewords. Thus, the statement in the Office Action that, "there would be no need to add an extra bit" is an incorrect assumption that ignores the context of the algorithm.

C. **"WHY MUST ANY TWO N-BIT WORDS DIFFER IN AT LEAST ONE BIT?"**

Looking at page 7, lines 11-13, the Examiner questioned why must any two n-bit words differ in at least one bit.

Perhaps the Examiner is thinking of two consecutive n-bit words occurring in a data-stream and wondering why that should be? If we split the data stream into consecutive n-bit words, no two consecutive n-bit words need be different, nor any two non-consecutive words for that matter. They can be identical. But, that is not what we were describing in line 11 of page 7.

Page 7 lines 11-13 state, "Also, since any two n-bit words differ in at least one bit, their corresponding (n+1)-bit code words also differ in at least one bit, which ensures that the complements of these code words are also different." This follows a discussion on page 7, lines 7-9 discussing that each n-bit data word is assigned to a pair of unique (n+1)-bit code words.

The cited paragraph is therefore describing that each unique n-bit data word pattern has a unique code word pattern. In other words, there are enough unique code word patterns to implement the code, thus allowing the code to be reliably decoded. The cited paragraph does not relate to whether successive, same data words can be encoded by the encoder, which they can.

Basically, the algorithm performs encode-decode mapping in a reversible way. That is, there will not be any ambiguity in decoding a given (n+1)-bit word: a given (n+1)-bit code word will decode into a unique n-bit data word.

D. **"COULDN'T DUPLICATE WORDS OCCUR? AS APPLICANT STATES ON LINE 18 OF PAGE 7, REPETITION OF THE SAME WORD RESULTS IN AN UNBOUNDED RDS"**

In the last sentence of paragraph 2 on page 5 of the Office Action, the Examiner refers to page 7, line 18 of the specification in which states that repetition of the same word results in an unbounded RDS.

Yes, duplicate words can occur. But algorithm works in such a way as to bound the RDS of the encoded sequence even though the RDS in the data sequence could grow unboundedly as the

Examiner stated. Page 7, line 18 relates to the uncoded n-bit words b. Even if the incoming data word string has an unbounded cumulative RDS, the cumulative RDS of coded word sequence c is bounded.

Suppose that the accumulated RDS at the present time in the coded sequence equals R, and suppose that a particular data word w keeps repeating itself thereafter. If the running-digital-sum of w equals r, and if R and r both have the same sign, then we complement bits in w and append a "1" to it, so its running-digital-sum equals (-r + 1) and the total accumulated RDS now is updated to $R_{new} = R - r + 1$ which is less than R. Now, if w occurs again, we will again check to see if r and R_{new} have the same sign, and if so, we will flip w and append a "1" to it as before, so the new accumulated RDS will be ($R_{new} - r + 1$) which will be less than R_{new}. This is how the accumulated RDS is prevented from growing even if duplicates occur in the data stream.

E. **"HOW DOES APPLICANT ESTABLISH A CYCLICAL RDS PATTERN?"**

The Examiner questions how a "tandem" combination of an n-bit pattern and its complement as stated on page 7, lines 20-21 produces an oscillating RDS pattern.

It is the "RDS of the encoded sequence" (page 7, line 22) that has the behavior recited in lines 22-23, not the incoming data stream. When the Examiner suggests that the complement would have the same RDS magnitude but an opposite sign such that a combination of RDS values having opposite values would result in zero RDS, the Examiner appears to be overlooking the fact that the encoder not only selectively complements the incoming tandem data blocks but also adds an additional bit (1 or 0) to form the code word. Whether a particular block gets complemented and whether a 1 or a 0 gets appended to that block depends on the current cumulative RDS of the encoded sequence.

This operation has a qualitative effect on the RDS of the overall sequence as new data words are being encoded into code words. Take the following example, which was generated using the algorithm shown in FIG. 2, where Ri is the RDS of a data word, CW is the resulting code word, Rcw is the RDS of that code word and Rseq is the cumulative RDS of the code word sequence:

Data Words	000	111	000	111	000	...
Ri	-3	3	-3	3	-3	...
CW	0000	1110	1111	0001	0000	...
Rcw	-4	2	4	-2	-4	...
Rseq (initially 0)	-4	-2	2	0	-4	...

The RDS of the encoded sequence is clearly oscillating. It increases, crosses zero and changes direction. This example therefore answers the Examiner's question, "How could this possibly occur?"

F. **TABLES 1 AND 2**

The Examiner requested a clarification of Tables 1 and 2 and specifically the encoding of the 3-bit data blocks "010", "111", and "001". Applicants would like to remind the Examiner that each 3-bit data block in this example is not being encoded in isolation but rather is being encoded relative to the accumulated RDS of a running code word stream. Therefore, the actions performed by the encoder depend not only on the present 3-bit data block but also the accumulated RDS of the output stream.

With respect to the first occurrence of data block "010" in the first row of Table 2, and using the variables in FIG. 2, Ri=-1 and R=0 thus fitting the condition (R>=0 and Ri<0) such that a "0" is appended to the data block (without complementing the data word) to produce the corresponding code word "0100". Summing the RDS of this code word (-2) with the accumulated RDS of the code word sequence before encoding the present data block (0), the accumulated RDS of the code word stream after encoding becomes (-2).

When data block "111" is received, in the third row of Table 2, the accumulated RDS of the code word sequence is "0" (R<=0) and the RDS of the present data block is "3" (R>0). This satisfies the "ELSE IF" condition in FIG. 2 such that a 0 is appended to the data block to form the code word. Thus, Table 2 clearly explains the answer to the Examiner's question what happened?"

The Office Action further questions how a "1" is appended for encoding "001" to --1101--. This appears in row five of Table 2. At this stage, the RDS of the data block has the same sign (-1) as the accumulated RDS for the code word stream (-2), thus satisfying the condition (R<0 and Ri<=0) within the "IF" statement in FIG. 2. Thus, the data block is complemented and a "1" is appended to generate the code word --1101--.

Tables 1 and 2 therefore provide a detailed example of how the encoder would operate according to one embodiment of the present invention, which could easily be understood by a person of ordinary skill in the art.

On page 6 of the Office Action, the Examiner questions how a zero RDS value is treated and suggests that a zero value does not fit into "the sign of the RDS" limitations currently pending in the claims. Applicants respectfully disagree since, within a sequence of signed numbers, a zero can clearly be

interpreted as having a sign. The fact that +0 and -0 are the same does not mean that zero can have no sign. It only means that the sign makes no difference. Thus, a zero can fit into "the sign of the RDS" limitations currently present in the claims.

G. **FIGURE 4**

With respect to step 404, the Examiner has asked, "What happens if the number of bits is even resulting in unsigned RDS values?" The RDS values are signed. A zero can be treated as a positive RDS or as a negative RDS. (See, e.g. page 6, lines 17-20). It makes no difference. Within the method recited in claim 1, for example, a person of ordinary skill in the art could easily implement the method either way. This boundary detail is clearly within the understanding of a person of ordinary skill in this art, and is explicitly described in the specification.

IV. CONCLUSION

Applicants' attorney has addressed each question raised by the Examiner, and these questions appear based on a misunderstanding by the Examiner rather than a lack of enablement by the specification. In view of the above-discussion, the original specification clearly supports the presently pending claims, and the specification is sufficient to enable a person of ordinary skill in the art to make and use the invention that is being claimed without undue experimentation.

Applicants therefore respectfully request that the rejections of claims 1-16 under §112 be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: ARego

Alan G. Rego, Reg. No. 45,956
Suite 1400 - International Centre
900 Second Avenue South
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

DDB:AGR:tkj